Pixel
Semiconductor
A Cirrus Logic Company

Alpine *CDX*

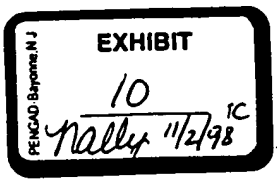
Confidential Information

Alpine CDx Design Specification

Version 1.1

03/03/94

John Schafer
(214) 985-2259



CL 41087

Table of Contents

CL 41088

1. Overview

The Alpine CDx is the first product in the Alpine family to integrate video playback support. Based on the 5430 GUI accelerator, the Alpine CDx allows high quality playback of video from CD ROMs or disk drives at a low cost premium over the existing 5430.

1.1. Detailed Pin Descriptions

(Insert from 5430 datasheet)

1.2. Functional Description and Block Diagram

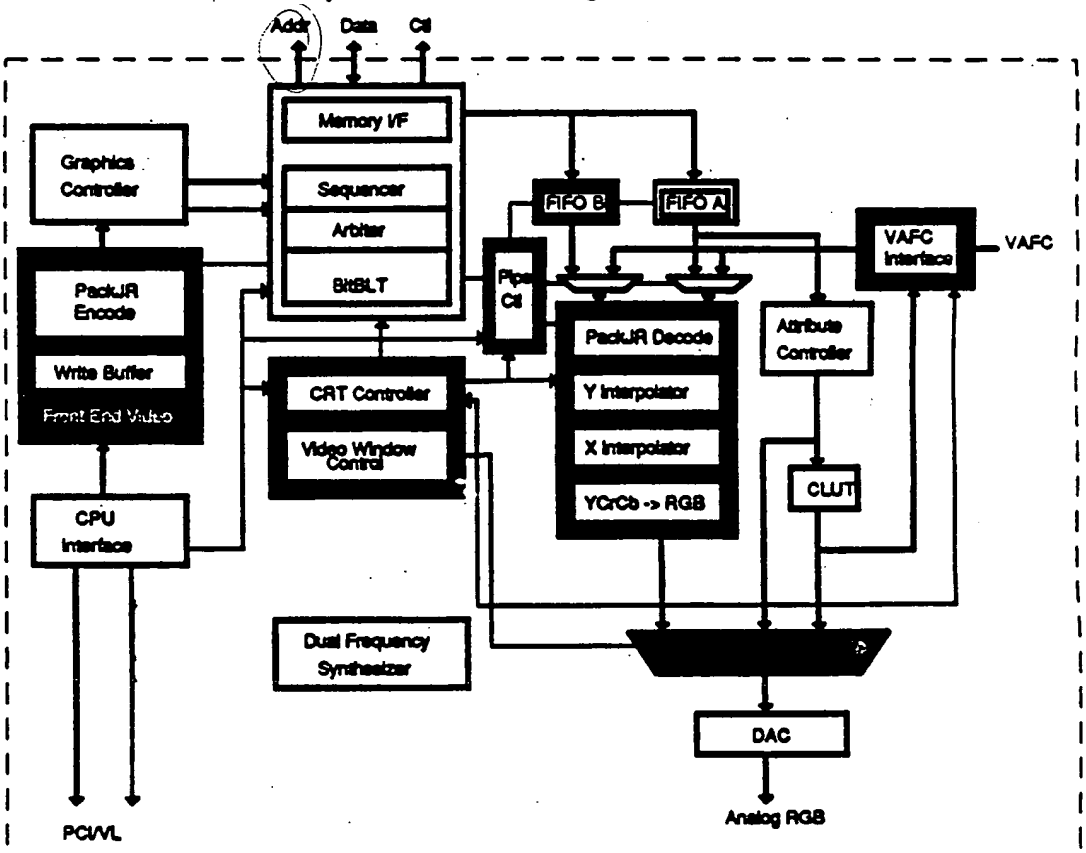


Figure 1. Alpine CDx Block Diagram

1.3. Timing Requirements

CL 41090

(Insert from 5430 datasheet with appropriate changes)

1.4. Cost Constraints

The cost goal for the Alpine CDx is a maximum 10% premium over the existing 5430. This amounts to 5000-6000 gates of additional logic. A die size estimate for the additional logic is included in appendix A.

1.5. Test Requirements

The confidence test developed for the 5430 will form the core test vector set to verify compatibility of the Alpine CDx. These confidence tests include IKOS waveform driven tests and Voyager VHDL driven tests. All tests added to verify the video acceleration and overlay features should use the Voyager VHDL approach.

In addition to the large number of confidence tests, a more streamlined set of fault grading and speed tests must be developed for manufacturing. Wherever possible, the existing 5430 fault tests will be used for the Alpine CDx. After the addition of new vectors for the video sections, the entire test vector set should be validated using the Zycad fault grader.

A checklist of required tests is included in appendix B.

1.6. Technology and Layout Considerations

The first production release of the Alpine CDx will use the Cirrus 0.8um, double metal CMOS process. Schematic capture and layout for new custom blocks will be performed in Plano using Pixel's datapath library. Additional datapath cells will be created as necessary. The layout will be generated using a hand place and route of the datapath cells. If possible, a hand place, autoroute of the datapath cells will be performed using Cell Ensemble.

Standard cell control logic used within the video pipeline will use the Pixel 0.8um library with the Cadence schematic environment. Place and route for these control blocks will be performed at Pixel using Cell Ensemble and be included in the layout of macrocell blocks delivered to Fremont for full chip routing.

All design changes made outside these macrocells should utilize the User Interface 0.8um standard cell library within the Mentor schematic environment. These cells will then be routed as part of the full chip layout development at Fremont.

In summary, all cells used within a custom macrocell will be developed using Pixel datapath and standard cell libraries within the Cadence environment. All other design changes will be made using the User Interface library within the Mentor environment.

A 0.6um double or triple metal process will most likely be used for future cost reductions of the Alpine CDx. The macrocell development for the 0.6um shrink should begin as soon as the 0.8um version has been completed.

2. Detailed Module Descriptions

2.1. Frontend Video Pipeline

2.1.1. Functional Description and Block Diagram

The frontend video pipeline and control supports PackJR encode and video display memory mapping. This block requires significant reverse engineering of the write buffer and graphics datapaths and control to ensure compatibility with the existing design. The preliminary plan calls for a separate address decode for video data to allow tagging of video data in the write buffer. Data read from the write buffer will be processed by the existing graphics path or the new video path based on the tag.

2.1.2. Video Data Path

2.1.2.1. Input/Output Definitions

2.1.2.2. Functional Description and Block Diagram

The video data path may optionally convert CCIR 601 format YCbCr to out proprietary PackJR format. PackJR is a modified YCbCr 411 format using 5 bits for each luminance component and 6 bits for each chrominance component. The 32 bit PackJR words are defined as follows:

D(31:27)	D(26:22)	D(21:17)	D(16:12)	D(11:6)	D(5:0)
Y3(4:0)	Y2(4:0)	Y1(4:0)	Y0(4:0)	U0(5:0)	V0(5:0)

2.1.2.3. Timing Requirements

(TBD)

2.1.2.4. Area Constraints

(TBD)

2.1.2.5. Test Requirements

(TBD)

2.1.2.6. Layout Considerations

This block will be implemented using the User Interface 0.8um standard cells within the Mentor environment.

2.2. Backend Video Pipeline

2.2.1. Overview

The backend video pipeline must perform format conversion, interpolated zooming, and color space conversion for a rectangular region of the display defined as the *Video Window.*

The video pipeline datapath, as shown in figure 1, consists of dual CRT FIFOs, a PackJR decode block, independent X and Y interpolating zoomers, and an YCbCr->RGB color space converter. The size and location of the video window is controlled by the Video Window Control block, which works in conjunction with the VGA CRT controller.

The dual CRT FIFOs are used to provide data from two adjacent scan lines for interpolative Y zooming. FIFO A is the standard VGA CRT FIFO which will store graphics data and the first line of video data. FIFO B is a new FIFO which stores the second line of video and a video window active tag generated by the Video Window Control Block.

An input mux to the PackJR decode allows selection of VAFC or Display Memory data. The PackJR decode block accepts two 32 bit words, one from each adjacent scan line. Each word represents four YCbCr pixels, which are expanded and error diffused to provide four 16 bit YCbCr pixels. For video stored in standard 555 RGB, 565 RGB, or 16 bit YCbCr data formats, the PackJR decode block may be bypassed.

The Y interpolator accepts two vertically adjacent 16 bit RGB or YCbCr pixels from the PackJR block, and calculates a resampled output pixel using a four subpixel granularity. The X interpolator accepts horizontally adjacent pixels from the Y interpolator, and calculates the resampled output pixel using a four subpixel granularity. For line replication mode, the Y interpolator may be bypassed.

The Pipe Control block controls reading from the CRT FIFOs and generation of interpolation coefficients and timing control to provide continuous 1x to 4x zooming with a four pixel granularity.

The CRT and Video Window Control block generates memory addresses for screen refresh based on the size and position of the video window.

The Video Window Control Register block contains the new IO registers used to specify the color space, zoom factor, size and position of the video window.

2.2.2. CRT and Video Window Address Control (VWACTL)

2.2.2.1. Input/Output Definitions

Name	Type	Src/Dst Block	Delay	Description
LDB(7:0)	Input	HIF	tbd	Latched data bus for register writes.
CRTRS(19)	Input	REGIO	tbd	Register select for R13
CRTRS(15:12)	Input	REGIO	tbd	Register selects for R0F, R0E, R0D, and R0C.
VWRS(14:11)	Input	REGIO	tbd	Register select for VW Start Address and offset.
VWRS(7:4)	Input	REGIO	tbd	Register select for video window horizontal start and size.
IOWST1	Input	HIF	tbd	IO Write State 1
IORST	Input	HIF	tbd	IO Read State
W3X5	Input	REGIO	tbd	Write address decode for CRT data port.
R3X5	Input	REGIO	tbd	Read address decode for CRT data port.
R14(6)	Input	REGIO	Static	Double word mode.
R17(6:5)	Input	REGIO	Static	Address wrap. Works with R17B6 in following manner: R17B6 R17B5 Operation 0 0 Rotate 14 bit address left one position 0 1 Rotate 16 bit address left one position 1 X No rotation.
R17(1:0)	Input	REGIO	Static	R17-1 : Scan row counter select. If set to '0', row scan counter (1) is substituted for address bit (14) to provide Hercules™ compatibility. R17-0 : Substitute for address bit (14) for CGA compatibility.
R0A(5)	Input	REGIO	Static	Cursor disable.
R1A(1:0)	Input	REGIO	Static	R1A-0 : Interlace enable. R1A-1 : Start address double buffer enable.
R1B(4)	Input	REGIO	Static	Offset bit 8.

R1B(1)	Input	REGIO	Static	Extended address wrap enable. See reference manual for details.
LR1(3:2)	Input	REGIO	Static	Screen Start Address Bits 18:17.
LR1(0)	Input	REGIO	Static	Screen Start Address Bits 16.
SR7(0)	Input	SQRG	Static	High resolution color Mode. 1 character clock = 8 pixels.
LNCOM	Input	VT	cck->Q + 1 gate.	Indicates line count equal to line compare (Begin screen B).
ODD	Input	VT	cck->Q	Odd field indicator.
VRET1L	Input	VT	cck->Q + 3 gates	Vertical retrace.
VDISP	Input	VT	cck->Q + 1 gate	Vertical display active asserted on all active lines ?
VDIS1	Input	VT	cck->Q + 1 gate	Asserted on last active line.
VTVWACT	Input	VT	cck->Q	Video window active from VT block.
HRET1P	Input	HT	cck->Q + 2 gates.	Asserted for one character clock at start of horizontal retrace.
NEXTLINE	Input	ROWSC	cck->Q	Start of line indicator.
LSROW	Input	ROWSC	cck->Q	Asserted at start of new character row
ROW(1:0)	Input	ROWSC	cck->Q + 1 gate	Row count.
CHRINC	Input	SQRC	mck->Q + 1 gate.	Character increment
CALD	Input	SQSY	mck->Q + 1 gate.	Character address load. Controls loading of character address with SLA.
APA	Input	VIDCNT	Static	All points addressable. 0 = text mode, 1 = graphics mode
RSTn	Input	CRTC	cck->Q	CPU reset.
CCRST	Input	CRTC	cck->Q	CPU reset extended by on cck.
TR1(5:0)	Input	CRTC	Static	Test mode control bits.
LINEREP	Input	VPCTL	Static	Line replicate enable.
VWEN	Input	VPCTL	Static	Video window enable.
VIDMODE	Input	VPCTL	Static	Video mode: 0 = video on graphics, 1 = graphics on video.
NEWVIDLINE	Input	VPCTL	vpck->Q	New video line enable.
CCK	Input	SQV		Character clock.

MCK	Input			Memory clock.
IOD1N(7:0)	Output	HIF	tbd	Inverted IO output data bus.
MA(18:0)	Output	RSCSA	mck->Q + 4 gates	Memory address.
Eo8	Output	SQRC	mck->Q + 1 gate	End of 8.
EOPn	Output	SQRC	mck->Q + 1 gate	End of page.
CURS	Output	VIDFF	mck->Q + 4 gates	

2.2.2.2. Functional Description and Block Diagram

This block replaces the CHRADM and CHRADV blocks of the 5429 database. As shown in the block diagram, the VWACTL maintains three memory address pointers, the first for graphic data, the second for first line video data, and the third for second line video data. The graphics address pointer is used to transfer graphics data to CRT FIFO A in a similar manner as in the 5429. The first and second line addresses are used to transfer video data to CRT FIFOs A and B, respectively, to support vertical interpolation within a video window.

The block diagram consists of one 18-bit counter for each address pointer, and a shared 18 bit adder used to calculate the start of line addresses.

A key change from the 5429 is in the synchronization of the start of line address calculator and the address counter. In the 5429, the start of line address calculator is synchronous to the character clock, whereas the address counter is synchronous to the memory clock. In the Alpine CDx, both will be synchronous to the memory clock. The event used to load the start of line address is an mclk synchronized HRET1P, which is asserted at the start of the horizontal retrace interval.

The control block contains miscellaneous decode from the CHRADV and CHRADM blocks. as well as a new state machine to control the start of line accumulators and address counters. A counter within the control block is used to determine the transition points between graphics and video along a horizontal scan. The vertical transition points will be determined by comparators added to the Vertical Timing (VT) block.

2.2.2.2.1. IO Registers

Enable	Name	Description
CRTRS(12)	R0C(7:0)	Screen Start Address Bits 15:8.
CRTRS(13)	R0D(7:0)	Screen Start Address Bits 7:0
CRTRS(14)	R0E(7:0)	Cursor address bits 15:8
CRTRS(15)	R0F(7:0)	Cursor address bits 7:0.
VWRS(4)	VWGST	Video Window Horizontal Region 1 Size
VWRS(5)	VWHGSZ	Video Window Horizontal Region 2 Active Size
VWRS(6)	VWHVSZ	Video Window Horizontal Region 2 Skip Size
VWRS(7)	VWHOVR	Video Window Horizontal Overflow
VWRS(11)	VWASTL	Video Window Start Address Low Byte
VWRS(12)	VWASTM	Video Window Start Address Middle Byte
VWRS(13)	VWASTH	Video Window Start Address High Byte
VWRS(14)	VWOFF	Video Window Address Offset

In order to minimize routing, four control registers will be moved to the control block from REGIO. An additional eight control registers are used to specify the horizontal size and location of the video window, as well as the video start address and offset. Refer to the Alpine CDx functional specification for detailed register definitions. A summary of the local control registers is shown in the preceding table. The IO mapped control registers are accessed using the same method as the in the REGIO block. Each register bit consists of a D-type latch and an inverting tristate buffer. When the input signals W3X5, RS and IOWST1 are all asserted, data is loaded into the control register latches from the input data bus LDB(7:0). When R3X5, RS, and IORST

signals are all asserted, data are enabled from the latches onto the output data bus IOD1N(7:0) through inverting tristate buffers.

2.2.2.2.2. State machine.

The datapath shown in the following block diagram is controlled by a state machine within the control block, as described in the following paragraphs.

At the start of each frame, the Graphics Start of Line Address register (GRSLA) is loaded with the Graphics Screen Start Address (GRSTADDR). The First Line Start of Line Address (FLSLA) is loaded with the Video Start Address (VDSTADDR). The Second Line Start of Line Address (SLSLA) is loaded with VDSTADDR, then incremented by the video address offset (VDOFFSET).

During the horizontal retrace interval prior to each active line, the address counters are initialized with their corresponding start of line addresses. The Graphics Address Counter (GRACNT) is loaded with GRSLA. The First Line Address Counter (FLACNT) is loaded with FLSLA. The Second Line Address Counter (SLACNT) is loaded with SLSLA.

Each start of line address is then incremented by the appropriate offset. In order to conserve die area, a common adder is used for all start of line address plus offset calculations. The state machine must sequence through three accumulation cycles. If possible, two mclk cycles should be used for each add, to minimize the performance requirements of the 18 bit adder. In order to support line replication for variable zoom factors, the FLSLA is incremented only if the input signal NEWVIDLINE is asserted.

Once the address counters are initialized, the display memory to CRT FIFO transfers may begin under control of the sequencer. Each active horizontal scan line is divided into three sequential regions. For video on graphics mode, indicated by VIDMODE equal to zero, Region 1 is graphics data, Region 2 is video data, and Region 3 is graphics data. For graphics on video mode, indicated by VIDMODE equal to zero, Region 1 is video data, Region 2 is graphics data, and Region 3 is video data. Depending on the size and location of Region 2, Regions 1 or 3 may be of size zero.

Video on graphics mode:

At the start of each active line , the Regior. Counter (RGCNT) is loaded with the Region 1 size, specified in terms of graphics DWORDS by the Video Window Horizontal Region 1 Size (VWR1SZ). Graphics data is then transferred to CRT FIFO A from the display memory address specified by GRACNT for the duration of Region 1. For each DWORD transferred, GRACNT is incremented by one, and RGCNT is decremented by one.

At the end of Region 1, indicated by RGCNT equal to zero, the RGCNT is loaded with the Region 2 size, specified in terms of video DWORDS by the Video Window Horizontal Region 2 Active Size register (VWHR2ASZ). Video data is then transferred to CRT FIFO A from the display memory address specified by the FLACNT. If vertical interpolation is enabled, the sequencer must alternate between first line video transfers from display memory address FLACNT to CRT FIFO A, and second video line transfers from memory address SLACNT to CRT FIFO B. For each DWORD transferred to CRT FIFO A, FLACNT is incremented by one. For each DWORD transferred to CRT FIFO B, SLACNT is incremented by one and RGCNT is decremented by one.

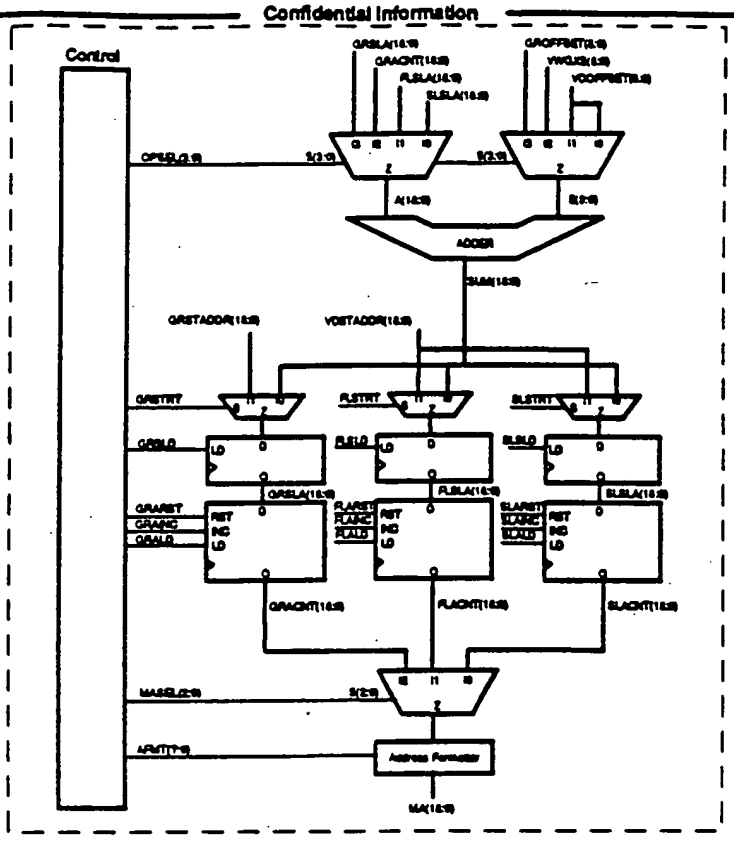
At the end of Region 2, indicated by RGCNT equal to zero, the GRACNT is incremented by the Video Window Region 2 Skip (WHR2SSZ), specified in terms of graphics DWORDS. This provides the graphics start address for Region 3. Graphics data is then transferred to CRT FIFO A from the display memory address specified by GRACNT. For each DWORD transferred, GRACNT is incremented by one. These transfers continue until the end of the active line.

Graphics on video mode:

At the start of each active line , the Region Counter (RGCNT) is loaded with the Region 1 size, specified in terms of video DWORDS by the Video Window Horizontal Region 1 Size (VWR1SZ). Video data is then transferred to CRT FIFO A from the display memory address specified by the FLACNT. If vertical interpolation is enabled, the sequencer must alternate between first line video transfers from display memory address FLACNT to CRT FIFO A, and second video line transfers from memory address SLACNT to CRT FIFO B. For each DWORD transferred to CRT FIFO A, FLACNT is incremented by one. For each DWORD transferred to CRT FIFO B, SLACNT is incremented by one and RGCNT is decremented by one.

At the end of Region 1, indicated by RGCNT equal to zero, the RGCNT is loaded with the Region 2 size, specified in terms of graphics DWORDS by the Video Window Horizontal Region 2 Active Size register (VWHR2ASZ). Graphics data is then transferred to CRT FIFO A from the display memory address specified by GRACNT for the duration of Region 2. For each DWORD transferred, GRACNT is incremented by one, and RGCNT is decremented by one.

At the end of Region 2, indicated by RGCNT equal to zero, the FLACNT and SLACNT are incremented by the Video Window Region 2 Skip Size (WHR2SSZ), specified in terms of video DWORDS. This provides the video start addresses for Region 3. Video data is then transferred to CRT FIFO A from the display memory address specified by the FLACNT. If vertical interpolation is enabled, the sequencer must alternate between first line video transfers from display memory address FLACNT to CRT FIFO A, and second video line transfers from display memory address SLACNT to CRT FIFO B. For each DWORD transferred to CRT FIFO A, FLACNT is incremented by one. For each DWORD transferred to CRT FIFO B, SLACNT is incremented by one. These transfers continue until the end of the active line.



2.2.2.2.3. Graphics Address Formatter

To maintain graphics mode compatibility, the GRACN must be reformatted to form the graphics memory address (GRMA). Refer to CHRADM specification in Appendix C for a detailed description of this function.

2.2.2.3. Timing Requirements

This block runs at the maximum MCLK frequency of 60 MHz. Special attention must be taken to synchronize all control signals to MCLK.

2.2.2.4. Area Constraints

Refer to appendix A for module die size estimate.

2.2.2.5. Test Requirements

The test modes within the CHRADM and CHRADV blocks of the 5429 must be supported to allow test vector compatibility. Additional test modes may be defined to control loading and incrementing of video address and region counters. A checklist of simulations required is included in appendix B.

2.2.2.6. Layout Considerations

Due to schedule constraints, this block will most likely be implemented using the Pixel standard cell library.

2.2.3. Video Pipe Control

2.2.3.1. Input/Output Definitions

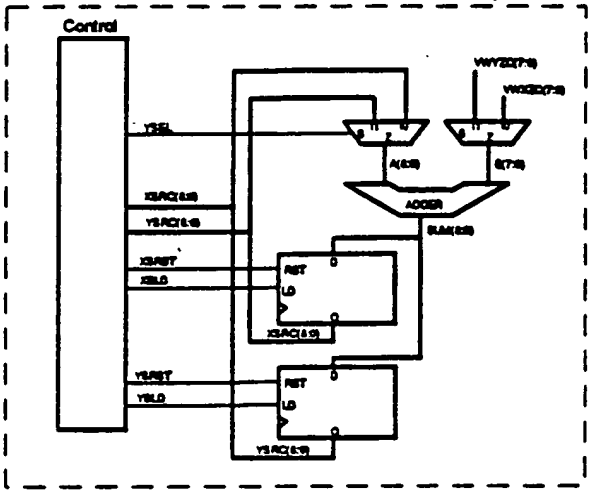
Name	Type	Src/Dst Block	Delay	Description
LDB(7:0)	Input	HIF	tbd	Latched data bus for register writes.
VWRS(3:1)	Input	REGIO	tbd	Register select for MCR,XZC, and YZC control registers.
IOWST1	Input	HIF	tbd	IO Write State 1
IORST	Input	HIF	tbd	IO Read State
W3X5	Input	REGIO	tbd	Write address decode for CRT data port.
R3X5	Input	REGIO	tbd	Read address decode for CRT data port.
VWACTOUT	Input	FIFOB	tbd	Video window active flag out.
VPCLK	Input	tbd	tbd	Video pipeline clock.
LINEREP	Output	VPACTL	Static	Line replicate enable.
VWEN	Output	VPACTL	Static	Video window enable.
VIDMODE	Output	VPACTL	Static	Video mode: 0 = video on graphics, 1 = graphics on video.
NEWVIDLINE	Output	VPACTL	vpck->Q	New video line enable.
IOD1N(7:0)	Output	HIF	tbd	Inverted IO output data bus.
FBRDEN	Output	FIFOB	tbd	FIFO B read enable.
PJRBLK	Ouput	VWPJR	vpclk->Q	PackJR blank enable.
VAFCEN	Output	VWPJR	vpclk->Q	VAFC enable
PJRRAND(2:0)	Ouput	VWPJR	Static	Random data for error diffusion.
PJRSEL(3:0)	Output	VWPJR	vpclk->Q	PackJR Select
PJRCLKEN(1:0)	Output	VWPJR	tbd	PackJR Clock Enable
XYFMT(3:0)	Output	VWXY	Static	XY Interpolator Color Space Format.
YCOEF(3:0)	Ouput	VWXY	Static	Y Interpolation Coeficient
YBYPASS	Output	VWXY	Static	Y Interpolator bypass enable.
XCOEF(3:0)	Output	VWXY	vpclk->Q	X Interpolation Coeficient
XBYPASS	Output	VWXY	vpclk->Q	X Interpolator bypass enable.
XYCLKEN	Output	VWXY	vpclk->Q	XY Interpolator clock enable.
CSCBYP(1:0)	Ouput	VWXY	Static	Color space converter bypass enable.
CSCCPAK	Output	VWXY	Static	Cinepak conversion enable.
CSCCLKEN	Output	VWPCTL	tbd	Pipeline clock enable.

2.2.3.2. IO Register Definitions:

Enable	Name	Description
VWRS(1)	VWMCR	Video Window Master Control
VWRS(2)	VWXZC	Video Window Horizontal Zoom Code
VWRS(3)	VWYZC	Video Window Vertical Zoom Code

2.2.3.3. Functional Description and Block Diagram

The Pipe Control block controls reading from the CRT FIFOs and generation of interpolation coefficients and timing control to the video pipeline datapath blocks.



The zoom controls borrow heavily from the display window zoom control within the Px2070. In order to conserve die area, a common adder is shared by the X and Y Zoom Controllers.

Vertical Zoom

The 8 bit vertical zoom code (VW_VZC) used for interpolated or replicated resizing of the video window. The relationship between the Video Window Display Height and Source Image Height is determined as follows:

Display Height = truncation ((256/VW_VZC) * (Source Image Height)).

For example, a vertical zoom code of 128 would result in 2x vertical zoom operation, whereas a code of 64 would result in a 4x vertical zoom operation. Vertical zoom codes between 255 and 64 provide continuous zooming from 1x to 4x. A zoom code of zero indicates 1x scale.

The vertical zoom line selection and coefficient generation is controlled by the Y Source Address (YSRC), which is a binary number with one integer and 8 fraction bits. At the start of each frame the YSRC is reset to zero. At the beginning of each new active display line, the YSRC is incremented by the Video Window Y Zoom Code.

For interpolation mode, if the new integer portion of YSRC (YSRC-8), is not equal to the integer portion of the previous YSRC (LAST_YSRC-8), NEWVIDLINE is asserted to indicate to the memory address controller that a new video line must be fetched.

For line replication mode, NEWVIDLINE is asserted when the integer portion of YSRC+1/2, is not equal to LAST_YSRC-8. This operation is a round to nearest line, or nearest neighbor selection.

The Y interpolation coefficients determine the weighting factors for interpolation between two source lines to produce a destination line. The YCOEF and YBYPASS signals are determined from the two most significant fractional bits of YSRC as shown in the following table.

YSRC(7:6)	YCOEF(3:0)	YBYPASS	Operation
00	XXXX	1	Output = FL
01	0111	0	Output = (3/4 * FL) + (1/4 * SL)
10	1010	0	Output = (1/2 * FL) + (1/2 * SL)
11	1101	0	Output = (1/4 * FL) + (3/4 * SL)

FL and SL are the first and second lines in the vertical scan.

2.2.3.4. Horizontal Zoom

The 8 bit horizontal zoom code (VW_HZC) used for interpolated resizing of the video window. The relationship between the Video Window Display Width and Source Image Width is determined as follows:

Display Width = truncation ((256/VW_HZC) * (Source Image Width)).

For example, a horizontal zoom code of 128 would result in 2x horizontal zoom operation, whereas a code of 64 would result in a 4x zoom operation. Horizontal zoom codes between 255 and 64 provide continuous zooming from 1x to 4x. A zoom code of zero indicates 1x scale.

The horizontal pixel selection and coefficient generation is controlled by the X Source Address (XSRC), which is a binary number with one integer and 8 fraction bits. At the start of each line, during the horizontal retrace interval, XSRC is reset to zero. For every video pipeline clock within the active portion of the video window, the XSRC is incremented by the Video Window X Zoom Code.

If the new integer portion of XSRC (XSRC-8), is not equal to the integer portion of the previous XSRC (LAST_XSRC-8), a new pixel is enabled. This condition controls the advancing of video data through the pipeline and reading of video data from the CRT FIFOs.

The X interpolation coefficients determine the weighting factors for interpolation between two adjacent source pixels to produce a destination pixel. The XCOEF and XBYPASS signals are determined from the two most significant fractional bits of XSRC as shown in the following table.

XSRC(7:6)	XCOEF(3:0)	XBYPASS	Operation
00	XXXX	1	Output = FP
01	0111	0	Output = (3/4 * FP) + (1/4 * SP)
10	1010	0	Output = (1/2 * FP) + (1/2 * SP)
11	1101	0	Output = (1/4 * FP) + (3/4 * SP)

FP and SP are the first and second pixels along a horizontal scan.

PackJR

The random number generator for PackJR error diffusion is based on "primitive polynomial modulo 2" as described in chapter 7.4 of Numerical Recipes in C. The hardware implementation consists of an n-bit shift register and exclusive-or gates. The bit pattern is repeated 2^n-1 bits

where n is the length of the shift register. Some software simulations should be run to determine the qualitative effects of different shift register lengths. At the beginning of each new video frame, the shift register is loaded with a hard coded seed value. When error diffusion is disabled, the random number should be forced to zero by resetting the shift register. More details will be added here after the software analysis is completed.

2.2.3.5. Timing Requirements

This block runs at a maximum clock rate of 85 MHz. The 8 bit add for the X Source calculation must be performed in a single cycle and should be verified using Powermill.

2.2.3.6. Area Constraints

Refer to appendix A for module die size estimate.

2.2.3.7. Test Requirements

In order to simplify testing of this block, test logic must be added to the load and increment the X and Y Source registers. A checklist of simulations required is included in appendix B.

2.2.3.8. Layout Considerations

Due to schedule constraints, this block should be implemented using the Pixel standard cell library.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ BLACK BORDERS
- ☒ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.